

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-271391

(43)Date of publication of application : 26.09.2003

(51)Int.Cl.

G06F 9/445
G06F 12/06

(21)Application number : 2003-060467 (71)Applicant : SAMSUNG ELECTRONICS CO LTD

(22)Date of filing : 06.03.2003 (72)Inventor : LEE SEOK-HEON
LEE JAE-YOUNG
CHOI YOUNG-JOON
KWON SEOK-CHEON

(30)Priority

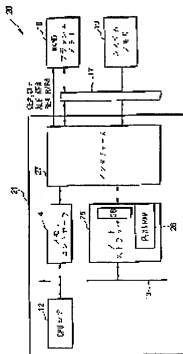
Priority number : 2002 200277291 Priority date : 06.12.2002 Priority country : KR
2002 200215856 23.03.2002 KR
2002 200212356 08.03.2002 KR

(54) BOOT SYSTEM USING NAND FLASH MEMORY AND ITS METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To use a NAND flash memory.

SOLUTION: This system having a CPU core, a system memory, and an interface for data communication between the CPU core and the system memory is provided with a NAND flash memory for storing at least a boot code, a boot strapper for controlling control signal transmission to the NAND flash memory, and for receiving the boot code from the NAND flash memory through the interface, and a RAM for storing the boot code accessed from the NAND flash memory. The system booting operation includes an operation for reading the boot code from the RAM by the CPU core.



[Date of request for examination] 18.01.2006

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.*** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] In the system which has an interface for the data communication between a core based CPU, a system memory, said core based CPU, and said system memory Control signal transmission to the NAND flash memory which stores a boot code at least, and said NAND flash memory is controlled. The bootstrapper which accepts the boot code from said NAND flash memory through said interface, System BUTINGU actuation is a system characterized by including the actuation which reads said boot code from said RAM by said core based CPU including RAM which stores said boot code accessed from said NAND flash memory.

[Claim 2] Said boot code is a system according to claim 1 characterized by including a system initialize program and a copy instruction program.

[Claim 3] Said bootstrapper is a system according to claim 1 characterized by including a NAND interface logic in order to carry out the interface of said NAND flash memory.

[Claim 4] Said bootstrapper is a system according to claim 1 characterized by including RAM for storing said boot code.

[Claim 5] Said RAM is a system according to claim 1 characterized by being located in the exterior of said bootstrapper.

[Claim 6] The system according to claim 1 characterized by including further the delay which delays actuation of said core based CPU until said boot code is stored in said RAM.

[Claim 7] Said delay is a system according to claim 6 characterized by realizing by said bootstrapper.

[Claim 8] Said interface is a system according to claim 1 characterized by realizing by the memory controller.

[Claim 9] Said memory controller is a system according to claim 8 characterized by including a NAND interface logic.

[Claim 10] Said bootstrapper is a system according to claim 9 characterized by including said RAM for storing said boot code.

[Claim 11] Said RAM is a system according to claim 9 characterized by being located in the exterior of said bootstrapper.

[Claim 12] Said RAM is a system according to claim 1 characterized by being the cache memory connected with said core based CPU through the 1st local bus.

[Claim 13] The system according to claim 12 characterized by including further the trumpet for connecting said bootstrapper with said cache memory.

[Claim 14] The system according to claim 12 characterized by including further the 2nd local bus for connecting said bootstrapper with said cache memory directly.

[Claim 15] Said interface is a system according to claim 14 characterized by realizing by the memory controller.

[Claim 16] Said memory controller is a system according to claim 14 characterized by including a NAND interface logic.

[Claim 17] Said RAM is a system according to claim 1 characterized by being the cache memory connected with said core based CPU through the local bus including said RAM for said bootstrapper to store said boot code.

[Claim 18] Said interface is a system according to claim 1 characterized by including the means for preventing that a system bus is simultaneously accessed by said memory controller and said bootstrapper, including further the memory controller for controlling said NAND flash memory and said system through said interface and system bus.

[Claim 19] Said memory controller is a system according to claim 18 characterized by including a NAND interface logic.

[Claim 20] The system characterized by including the NAND flash memory which stores OS program at least, and RAM which stores the code by which the data stored in said NAND flash memory are copied to said system memory in the system which has an interface for the data communication between a core based CPU, a system memory, said core based CPU, and said said system memory.

[Claim 21] It is the system according to claim 20 which said OS program is copied to said system memory at the time of initialization, and is characterized by said core based CPU performing said OS program by accessing said system memory.

[Claim 22] Said interface is a system according to claim 21 characterized by including a means which carries out thing prevention by which said system bus is simultaneously accessed by said memory controller and bootstrapper, including further the memory controller which controls said NAND flash memory and said system through said interface and system bus.

[Claim 23] Said memory controller is a system according to claim 22 characterized by including a NAND interface logic.

[Claim 24] In the system which has an interface for the data communication between a core based CPU, a system memory, said core based CPU, and said system memory It connects with the NAND flash memory which stores a boot code at least through a system bus at said NAND flash memory. The bootstrapper which accepts said boot code from said NAND flash memory is included. System BUTINGU actuation is a system according to claim 22 characterized by including the actuation which reads said boot code from said RAM by said core based CPU including RAM for said bootstrapper to store said boot code.

[Claim 25] Said interface is a system according to claim 24 characterized by including a means to prevent that said system bus is simultaneously accessed by said memory controller and bootstrapper, including further the memory controller which controls said NAND flash memory and said system through said interface and system bus.

[Claim 26] Said memory controller is a system according to claim 24 characterized by including a NAND interface logic.

[Claim 27] Said core based CPU is a system characterized by performing said OS program by accessing said system memory including the NAND flash memory which stores OS program at least in the system which has an interface for the data communication between a core based CPU, a system memory, said core based CPU, and said system memory, and the copy logical circuit to which said OS program is copied by said system memory at the time of the input of a system initialization signal.

[Claim 28] Said interface is a system according to claim 27 characterized by including a means to prevent that said system bus is simultaneously accessed by said memory controller and bootstrapper, including further the memory controller which controls said NAND flash memory and said system through said interface and system bus.

[Claim 29] Said memory controller is a system according to claim 28 characterized by including a NAND interface logic.

[Claim 30] In the system which has an interface for the data communication between a core based CPU, a system memory, said core based CPU, and said system memory The flash memory which stores a boot code at least, and the bootstrapper which controls control signal transmission to said flash memory, and accepts said boot code from said flash memory through said interface. The system characterized by including the 1st and 2nd memory controller which controls said flash memory selectively, and the selector which chooses one actuation among said 1st and 2nd memory controllers according to the class of flash memory.

[Claim 31] Said flash memory is a system according to claim 30 characterized by being one of NOR and NAND flash memories.

[Claim 32] For said 2nd memory controller, said 1st memory controller is a system according to claim 31 characterized by including a NAND interface logic including a NOR interface logic.

[Claim 33] The system according to claim 30 characterized by including further RAM which stores said boot code inputted from said flash memory.

[Claim 34] Said selector is a system according to claim 30 characterized by including the selection pin for choosing one actuation among said 1st and 2nd memory controllers.

[Claim 35] The approach characterized by including the phase of storing a boot code in a NAND flash memory beforehand, the phase of accepting a system initialization signal, the phase of transmitting said boot code to RAM from said NAND flash memory, and the phase of being made to perform said boot code in which said core based CPU was stored in said RAM, in the approach of driving the computing device which has a core based CPU, a system memory, and a memory controller.

[Claim 36] The approach according to claim 35 characterized by including further the phase of making activation of said core based CPU holding until the phase of transmitting said boot code to said RAM from said NAND flash memory at the time of the input of said system initialization signal is completed.

[Claim 37] The approach characterized by including the phase of setting initialization PARAMITTO as at least one connection pin, the phase of accepting a system initialization signal, the phase of transmitting OS program to said system memory from a NAND flash memory, and the phase where said core based CPU is made to perform said OS program from said system memory, in the approach of driving the computing device which has a core based CPU, a system memory, and a memory controller.

[Claim 38] The approach according to claim 37 characterized by including further the phase of setting up the mode register in said system memory using initialization PARAMITTO from said at least one connection pin before performing said OS program.

[Claim 39] In the approach of driving the computing device which has a core based CPU, a system memory, and a memory controller The phase of storing OS program beforehand in a NAND flash memory, The phase of storing beforehand the data containing the control data for causing the data transmission from said NAND flash memory in ROM, The phase of accepting a system initialization signal, and the phase where OS program from said NAND flash memory is made to be transmitted to said system memory using said control data from said ROM, The approach characterized by including the phase where said core based CPU is made to perform said OS program from said system memory.

[Claim 40] The approach according to claim 39 characterized by including further the phase of making activation of said core based CPU holding until the phase of transmitting said OS program to said RAM from said NAND flash memory at the time of the input of the initialization signal of said system is completed.

[Translation done.]

* NOTICES *

JPO and NCIPJ are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.*** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention still more specifically relates to the system driven using NAND flash memory equipment, and its BUTINGU approach about the system [BUTINGU / system / a computing device].

[0002]

[Description of the Prior Art] In the common personal computer or common computing device like PDA, the program prepared in BIOS (Basic Input/Output System : BIOS) is performed, when a personal computer or a computing device turns on. Many initialization functions are performed by performing a BIOS program. The function in which such a function generally checks the CMOS setup to custom-made setting setting out (custom settings), The function which carries out loading of an interrupt handler and the device driver, The function to perform the power-on self-test (power-on-self-test:POST) to the component or peripheral device installed like the function which initializes a register and a device management, and a disk drive, The function which displays system construction, the function to decide what kind of component can be driven, and the function which starts a bootstrapper sequence are included. Usually, a BIOS (or BUTINGU) program is stored in ROM, EPROM, or a NOR flash memory.

[0003] If a BUTINGUPURO gram is stored in ROM, since ROM is nonvolatile, the stored program cannot be changed. Even fine modification of the stored program makes exchange of ROM the need. When a BUTINGUPURO gram is stored in EPROM, in order to change the stored program, the program stored before must be eliminated. EPROM elimination requires a separate component or equipment further. If a BUTINGUPURO gram is stored in ROM or EPROM, any modification or updating which are required of a BUTINGUPURO gram cannot be performed easily, either. The stored program can be eliminated or updated when a BUTINGUPURO gram is stored in a NOR flash memory. However, if it compares with a NAND flash memory, a NOR flash memory has still larger magnitude to the given memory storage capacity, and its costs of manufacture are more high.

[0004] An example of a system which has the NAND flash memory which stores BIOS shows drawing 1 . This is U.S. Patent It is indicated by No.5,535,357. If drawing 1 is referred to, a system 10 contains the controller 11 for controlling the compound chip 16 and the compound chip 16 with which the system bus 17, the NAND flash memory 18, and the internal interface block 15 were included, and a system memory 19. A controller 11 is a central processing unit CPU, and this has the internal system bus 13 in the core based CPU 12 which performs a computing function, the memory controller 14, and a controller 11. The memory controller 14 uses the internal interface block 15 for carrying out the interface of the function which performs the memory map between the NAND flash memory 18 and a system memory 19, and is performed according to a memory map. The internal interface block 15 transmits the data which stored temporarily the data of the NAND flash memory in a register or storage equipment like RAM, and were temporarily stored through the system bus 17 in the bottom of control of the memory controller 14 to a system memory 19.

[0005] The internal interface block 15 includes the NOR interface logic 29 for carrying out an

interface to a system memory 19 or the memory controller 14 through the NAND interface logic 28 and system bus 17 for carrying out an interface to the NAND flash memory 18. The NOR interface logic 29 is a circuit generally used, in order to carry out an interface between a NOR flash memory, a memory controller, and/or a system memory. If a flash memory is a NOR flash memory, in order to make the signal from a flash memory in agreement with a NOR interface method (this contractor calls it a "ROM interface method"), the NAND interface logic 28 is unnecessary. A NOR interface method is a data transmission method in which the random access of memory is possible according to the address of a cutting tool/word unit. On the other hand, as for the data transmission by the NAND interface method, the data of the block unit instead of random access are transmitted by the block address and the command.

[0006] A system 10 transmits the data of the NAND flash memory 18 to the internal interface block 15 through the NAND interface logic 28 using a NAND interface method, and transmits said data to a system memory 19 through the NOR interface logic 29 using a NOR interface method. Since it must go via a NAND interface method and a NOR interface method in order to access data from the NAND flash memory 18, a data-access rate falls. Furthermore, the system performance of such equipment cannot be optimized. The reason is that time amount required to access BUTINGUKODO stored in the flash memory by the memory controller becomes one measuring element of system performance.

[0007] Furthermore, since the logical circuit which all the data of the NAND flash memory 18 must be loaded to the internal interface block 15, and all supports NAND and a NOR interface is required, the magnitude of the internal interface block 15 must be large. Therefore, in the case of the system 10 as shown in drawing 1, costs increase and the engine performance falls.

[0008]

[Problem(s) to be Solved by the Invention] The 1st object of this invention is offering the system which has the engine performance of a system which improved.

[0009] The 2nd object of this invention is offering the system which can lower manufacture costs.

[0010] The 3rd object of this invention is offering the system BUTINGU approach which can raise the engine performance of a system.

[0011] The 4th object of this invention is offering the system BUTINGU approach which can lower manufacture costs.

[0012]

[Means for Solving the Problem] The NAND flash memory in which the system which has an interface for the data communication between a core based CPU, a system memory, said core based CPU, and said system memory stores a boot code at least, The bootstrapper which controls control signal transmission to said NAND flash memory, and accepts the boot code from said NAND flash memory through said interface, System BUTINGU actuation includes the actuation which reads said boot code from said RAM by said core based CPU including RAM which stores said boot code accessed from said NAND flash memory. Desirably, said boot code includes a system initialize program and a copy instruction program.

[0013] According to 1 operation gestalt of this invention, RAM for storing said boot code is accumulated into said bootstrapper including a NAND interface logic for said bootstrapper to carry out the interface of said NAND flash memory.

[0014] Said RAM is located in the exterior of said bootstrapper in other operation gestalten.

[0015] Said system contains further the delay which delays actuation of said core based CPU until said boot code is stored in said RAM. Said delay is realized by said bootstrapper and said interface is realized by the memory controller. Said memory controller includes the NAND interface logic for carrying out the interface of said NAND flash memory by the NAND interface method.

[0016] According to other operation gestalten of this invention, said RAM is the cache memory connected with said core based CPU through the 1st local bus. A system contains further the trumpet for connecting said bootstrapper with said cache memory. The 2nd local bus connects said bootstrapper with said cache memory directly.

[0017] In order that said system may control said NAND flash memory and said system through

said interface and system bus, said interface includes the means for preventing that a system bus is simultaneously accessed by said memory controller and said bootstrapper, including a memory controller further.

[0018] According to other operation gestalten of this invention, the system which has an interface for the data communication between a core based CPU, a system memory, said core based CPU, and said system memory contains the NAND flash memory which stores OS program at least, and RAM which stores the code by which the data stored in said NAND flash memory are copied to said system memory. Desirably, said OS program is copied to said system memory at the time of initialization, and said core based CPU performs said OS program by accessing said system memory.

[0019] Moreover, the NAND flash memory in which the system which has an interface for the data communication between a core based CPU, a system memory, said core based CPU, and said system memory stores a boot code at least according to other operation gestalten, Connect with said NAND flash memory through a system bus, and the bootstrapper which accepts said boot code from said NAND flash memory is included. In order that said bootstrapper may store said boot code, system BUTINGU actuation includes the actuation which reads said boot code from said RAM by said core based CPU including RAM.

[0020] The NAND flash memory in which the system which has an interface for the data communication between a core based CPU, a system memory, said core based CPU, and said system memory stores OS program at least in other descriptions of this invention, At the time of two or more connection pins for setting up an initialization parameter, and the input of a system initialization signal Said core based CPU by accessing said system memory including the copy locking circuit to which said OS program is copied by said system memory Performing said OS program, system BUTINGU actuation includes the actuation which reads a boot code from said RAM by said core based CPU.

[0021] The flash memory in which the system which has an interface for the data communication between a core based CPU, a system memory, said core based CPU, and a system memory stores a boot code at least in other descriptions of this invention again, The bootstrapper which controls control signal transmission to said flash memory, and accepts said boot code from said flash memory through said interface, The 1st and 2nd memory controller which controls said flash memory selectively, and the selector which chooses one actuation among said 1st and 2nd memory controllers according to the class of flash memory are included.

[0022] Said flash memory is one of NOR and NAND flash memories. In said 1st memory controller, said 2nd memory controller includes a NAND interface logic including a NOR interface logic.

[0023] As for said system, it is desirable to include further RAM which stores said boot code inputted from said flash memory. Said selector contains the selection pin for choosing one actuation among said 1st and 2nd memory controllers.

[0024] The method of driving a core based CPU, a system memory, and the computing device that has a memory controller includes the phase of storing a boot code in a NAND flash memory beforehand, the phase of accepting a system initialization signal, the phase of transmitting said boot code to RAM from said NAND flash memory, and the phase of performing said boot code in which said core based CPU was stored in said RAM.

[0025] Such an actuation approach includes further the phase of making activation of said core based CPU holding until the phase of transmitting said boot code to said RAM from said NAND flash memory at the time of the input of said system initialization signal is completed.

[0026]

[Embodiment of the Invention] The system using the NAND flash memory by the operation gestalt of this invention is shown in drawing 2.

[0027] If drawing 2 is referred to, computing system 20 contains a controller 21, a system bus 17, the NAND flash memory 18, and a system memory 19. Computing system 20 includes the indispensable configuration of the computing device of a system [BUTINGU / system / PDA, a palm computer, laptop, a personal computer, or a system initialize program (generally, it is called a boot code)].

[0028] A controller 21 includes a core based CPU 12, the internal system bus 13, the memory controller 14, bootstrapper 25, and an interface 27. It can realize to a single semiconductor chip, and a controller 21 controls and manages what kind of memory like the NAND flash memory 18 connected with the system memory 17, and a system memory 19 generally.

[0029] A system bus 17 is used for the data transmission of a controller 21, the NAND flash memory 18, and a system memory 19.

[0030] The NAND flash memory 18 can store the boot code for BUTINGU [a system 20], and can store further an operating system (henceforth OS) and other programs, or data. OS is MS-DOS (trademark) or WINDOWS (trademark). This is performed by the controller 21 for operating a system 20 after BUTINGU or an initialization process. The data stored in the NAND flash memory 18 may be a software code which shows how a demand of a user is based and an application program is used with the configuration code of user storage equipment.

[0031] Desirably, a system memory 19 is DRAM and this operates as main memory which stores data, an instruction, and such a thing.

[0032] A core based CPU 12 performs OS, application, and a program of operation. The internal system bus 13 transmits the data from a core based CPU 12 and it, and transmits data between the memory controller 14 and bootstrapper 25.

[0033] According to the operation gestalt of this invention, a boot code is first stored in a NAND flash memory before initialization. According to initialization actuation, the boot code stored in the NAND flash memory 18 is transmitted to the interior RAM 26 in bootstrapper 25. After transfer of a boot code is completed, the boot code containing the initialization code and copy loop-formation instruction code of a system is performed by the core based CPU 12. When a system initialization code is performed, a controller 21, a system memory 19, and a peripheral device are initialized. As for copy loop-formation instruction code, the data of OS or others with which it was stored are loaded to a system memory 19. The magnitude of other data loaded to the interior RAM 26 by OS or the time amount of arbitration from the NAND flash memory is decided according to the value specified in boot code, or is determined by the hardware logic in bootstrapper 25. If it compares with NOR and a NAND interface logic (refer to drawing 1), since bootstrapper 25 includes only one NAND interface logic, bootstrapper 25 has small magnitude as compared with the interface block 15 of drawing 1. Furthermore, since the interior RAM 26 stores only a boot code, bootstrapper 25 has a still smaller capacity, when comparing with the interface block 15 which stores all the data of the NAND flash memory 18. Therefore, the price of the system 20 by this invention is still lower than that of the system of drawing 1.

[0034] The memory controller 14 controls and manages memory actuation, and memory actuation includes the actuation which stores OS or data stored in the NAND flash memory 18 in a system memory 19 through a system bus 17, and the actuation which reads data from a system memory 19. In such actuation, the memory controller 14 performs memory-mapping actuation between the NAND flash memory 18 and a system memory 19.

[0035] Next, actuation of the system 20 shown in drawing 2 is explained. When a system 20 turns on, bootstrapper 25 accepts a system initialization signal (for example, a power-up signal and a system-reset signal), and transmits the boot code stored in the NAND flash memory 18 to a system memory 19. While a boot code is transmitted to the interior RAM 26, bootstrapper 25 generates the control signal for holding actuation of CPU12. After transmission of a boot code is completed, a control signal is reset, it activates and a core based CPU 12 performs postBUTINGU (post-booting operation) actuation.

[0036] Moreover, in order to hold actuation of a core based CPU 12, delay (not shown) is used in a controller 21. For example, a system initialization signal (for example, a power-up signal and a system-reset signal) is simultaneously impressed to the bootstrapper 25 and delay which were connected with the core based CPU 12. Until after a boot code is transmitted to the interior RAM 26 from the NAND flash memory 18, the time of concentration of a system initialization signal is delayed. Delay is realized by a delay circuit or software. Delay is set up so that you may make it delayed for the same time amount as time amount required for boot code transfer, or it for a long time a little.

[0037] By activation of a core based CPU 12, the boot code stored in the interior RAM 26 is

performed. The hardware of a system 20 is initialized by performing the system initialization code in a boot code. By performing copy loop-formation instruction code of a boot code, a core based CPU 12 reads the program or other data like OS which were stored in the NAND flash memory. Desirably, this is performed per page through an interface 27 and the NAND interface logic 28 of bootstrapper 25. Henceforth, a core based CPU 12 copies a program like the data by which reading appearance was carried out, or OS to a system memory 19 through the memory controller 14 and an interface 27. OS is performed after copy actuation is completed. Therefore, when a BUTINGU process is completed, a system 20 is driven by OS from a system memory 19. [0038] According to the operation gestalt of this invention, BUTINGU actuation performed is performed through a single NAND interface by copying the boot code stored in the NAND flash memory 18 to the interior RAM 26, and transmitting OS to a system memory 19 at the degree. As the system of drawing 1 showed, when comparing with a two-step interface method (namely, a NAND interface method and a NOR interface method), a BUTINGU rate can be realized more quickly.

[0039] The actuation which reads the boot code stored in the NAND flash memory by bootstrapper 25, and transmits the boot code by which reading appearance was carried out to the interior RAM 26 is explained to a detail below based on drawing 2 and drawing 3. Drawing 3 shows the timing chart when reading a boot code from the NAND flash memory by the desirable operation gestalt of this invention.

[0040] Bootstrapper 25 outputs the control signal (for example, CLE, ALE, CE, WE, RE, and R/B) for reading the boot code which answered the system initialization signal and was stored in the NAND flash memory. A read-out instruction is generated by '00h' with the addresses ad0, ad1, and ad2. A system initialization signal contains a power-up signal (generated at the time of power-up of a system), a system-reset signal, or a re-reset signal.

[0041] The instruction latch enable signal CLE is activated when a predetermined instruction is inputted into the NAND flash memory 18. Address latch enable signal ALE is activated when the predetermined address is inputted into the NAND flash memory 18.

[0042] The NAND flash memory 18 answers activation of instruction latch enable signal CLE and chip enable signal CE# and write-in enable signal WE#, and receives 00h of read-out instructions through a system bus 17. Here, "#" means an active low state.

[0043] The NAND flash memory 18 answers activation of address latch enable signal ALE and chip enable signal CE# and write-in enable signal WE#, and accepts the address through a system bus 17. The number of the generated addresses is set up according to a NAND flash memory address step selection signal.

[0044] The three-step address input operation of the NAND flash memory 18 being shown in drawing 3 according to the operation gestalt of this invention, but not being restricted to the address input process having shown drawing 3 can be easily understood to this contractor of this field. The address input process which consisted of many phases from three steps or it according to the NAND flash memory address step selection signal is used.

[0045] According to this invention, bootstrapper 25 generates 00h of read-out instructions for reading the boot code stored in the NAND flash memory 18, and generates the addresses ad0, ad1, and ad2 in the degree. According to the addresses ad0, ad1, and ad2, reading appearance of the stored boot code is carried out by 00h of read-out instructions per page. The data by which reading appearance was carried out are temporarily stored in the internal buffer (not shown) of the NAND flash memory 18.

[0046] The actuation which copies data (for example, boot code) to an internal buffer in the activation ('low') condition of control signal R/B# is completed. Copy actuation of data (boot code: here, D0, D1, D2, D3) is not completed in the deactivation ('yes') condition of control signal R/B#. The data D0, D1, D2, and D3 stored in the internal buffer at the time of activation of read-out enable signal RE# are transmitted to a system bus 17. Henceforth, the data D0, D1, D2, and D3 on a system bus 17 are transmitted to the interior RAM 26.

[0047] According to the desirable operation gestalt of this invention, bootstrapper 25 answers a system initialization signal, holds or delays actuation of a core based CPU 12, and outputs control signal CE#, CLE and ALE, WE#, RE#, and R/B# to the NAND flash memory 18 through

the NAND interface logic 28 and an interface 27 simultaneously. The data by which reading appearance was carried out from the NAND flash memory 18 answer a control signal like CE#, CLE and ALE, WE#, RE#, and R/B#, and are transmitted to bootstrapper 25 through the NAND interface logic 28 and an interface 27. Bootstrapper 25 stores a boot code in the interior RAM 26, and cancels the HOLD status of a core based CPU 12. Or a core based CPU 12 is delayed time still longer than time amount required to store a boot code in the interior RAM 26. Delay is the timer which can be adjusted and is **. Therefore, while a core based CPU 12 is maintained by the HOLD status, the boot code stored in the NAND flash memory 18 is transmitted to the interior RAM 26.

[0048] Henceforth, the hardware of a system 20 is initialized by performing the system initialization code in the boot code loaded to the interior RAM 26. As for a core based CPU 2, by performing copy loop-formation instruction code in a boot code, it is desirable to transmit OS to a system memory 19 from the NAND flash memory 18 through the memory controller 14 and an interface 27. It can prevent that multiplexing or the data selection circuitry (not shown) within an interface 27 carries out multiplexing of the access to a memory bus 17, consequently the memory controller 14 and bootstrapper 25 access a system bus 17 to it simultaneously.

[0049] After OS transmission to a system memory 19 is completed, OS from a system memory 19 is performed and a system 20 is driven by OS.

[0050] The 2nd operation gestalt of the boot system using the NAND flash memory by this invention is shown in drawing 4 . According to this operation gestalt, the interior RAM 33 is arranged to the exterior of bootstrapper 32. It is desirable to perform access to the interior RAM 33 through the internal system bus 13 from bootstrapper 32. Unlike it, the data transmission actuation from the NAND flash memory 18 is the same as previous explanation.

[0051] The 3rd operation gestalt of the boot system using the NAND flash memory by this invention is shown in drawing 5 .

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.*** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] The drawing in which the computing system which has a general NAND flash memory is shown.

[Drawing 2] The drawing in which the system using the NAND flash memory by the desirable operation gestalt of this invention is shown.

[Drawing 3] The timing chart for explaining the process in which the boot code from a flash memory is transmitted.

[Drawing 4] Other operation gestalten of the computing system by this invention.

[Drawing 5] Other operation gestalten of the computing system by this invention again.

[Drawing 6] Other operation gestalten of the computing system by this invention again.

[Drawing 7] Other operation gestalten of the computing system by this invention again.

[Drawing 8] Other operation gestalten of the computing system by this invention again.

[Drawing 9] Other operation gestalten of the computing system by this invention again.

[Drawing 10] Other operation gestalten of the computing system by this invention again.

[Drawing 11] Other operation gestalten of the computing system by this invention again.

[Drawing 12] Other operation gestalten of the computing system by this invention again.

[Drawing 13] Other operation gestalten of the computing system by this invention again.

[Drawing 14] Other operation gestalten of the computing system by this invention again.

[Drawing 15] It is the flow chart which shows the procedure which carries out loading of the boot code stored in the flash memory by this invention.

[Description of Notations]

12 Core Based CPU

13 Internal System Bus

14 Memory Controller

17 System Bus

18 NAND Flash Memory

19 System Memory

20 System

21 Controller

25 Bootstrapper

26 Interior RAM

27 Interface

28 NAND Interface Logic

[Translation done.]

(51) Int. Cl. ⁷	識別記号	F I	テーマコード (参考)
G06F 9/445		G06F 12/06	520 F 5B060
12/06	520	9/06	610 K 5B076

審査請求 未請求 請求項の数40 O L (全17頁)

- (21) 出願番号 特願2003-60467 (P 2003-60467)
- (22) 出願日 平成15年3月6日 (2003.3.6)
- (31) 優先権主張番号 2002-077291
- (32) 優先日 平成14年12月6日 (2002.12.6)
- (33) 優先権主張国 韓国 (K R)
- (31) 優先権主張番号 2002-015856
- (32) 優先日 平成14年3月23日 (2002.3.23)
- (33) 優先権主張国 韓国 (K R)
- (31) 優先権主張番号 2002-012356
- (32) 優先日 平成14年3月8日 (2002.3.8)
- (33) 優先権主張国 韓国 (K R)

- (71) 出願人 390019839
三星電子株式会社
大韓民国京畿道水原市八達区梅蔭洞416
- (72) 発明者 李 錫煥
大韓民国京畿道水原市八達区梅蔭洞 (番地なし) 梅蔭住公アパートメント516-206
- (72) 発明者 李 載榮
大韓民国京畿道龍仁市水枝邑豊徳川里1065
シンジントンタウン住公1団地102棟803号
- (74) 代理人 100064908
弁理士 志賀 正武 (外1名)

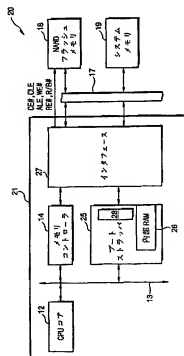
最終頁に続く

(54) 【発明の名称】 NANDフラッシュメモリを利用したブートシステム及びその方法

(57) 【要約】

【課題】 NANDフラッシュメモリを利用することを目的とする。

【解決手段】 CPUコア、システムメモリ、前記CPUコア及び前記システムメモリ間のデータ通信のためにインタフェースを有するシステムにおいて、少なくともブートコードを貯蔵するNANDフラッシュメモリと、前記NANDフラッシュメモリへの制御信号伝送を制御し、前記NANDフラッシュメモリからのブートコードを前記インタフェースを通じて受け入れるブートストラップと、前記NANDフラッシュメモリからアクセスされた前記ブートコードを貯蔵するRAMとを含み、システムブートアップ動作は前記CPUコアによって前記RAMから前記ブートコードを読み出す動作を含むことを特徴とするシステム。



【特許請求の範囲】

【請求項1】 CPUコア、システムメモリ、前記CPUコア及び前記システムメモリ間のデータ通信のためにインタフェースを有するシステムにおいて、少なくともブートコードを貯蔵するNANDフラッシュメモリと、前記NANDフラッシュメモリへの制御信号伝送を制御し、前記NANDフラッシュメモリからのブートコードを前記インタフェースを通じて受け入れるブートストラップと、前記NANDフラッシュメモリからアクセスされた前記ブートコードを貯蔵するRAMとを含み、システムブーティング動作は前記CPUコアによって前記RAMから前記ブートコードを読み出す動作を含むことを特徴とするシステム。

【請求項2】 前記ブートコードはシステム初期化プログラムとコピー命令プログラムを含むことを特徴とする請求項1に記載のシステム。

【請求項3】 前記ブートストラップは前記NANDフラッシュメモリをインタフェースするためにNANDインタフェースロジックを含むことを特徴とする請求項1に記載のシステム。

【請求項4】 前記ブートストラップは前記ブートコードを貯蔵するためのRAMを含むことを特徴とする請求項1に記載のシステム。

【請求項5】 前記RAMは前記ブートストラップの外部に位置することを特徴とする請求項1に記載のシステム。

【請求項6】 前記ブートコードが前記RAMに貯蔵されるまで前記CPUコアの動作を遅延させるディレーをさらに含むことを特徴とする請求項1に記載のシステム。

【請求項7】 前記ディレーは前記ブートストラップで実現されることを特徴とする請求項6に記載のシステム。

【請求項8】 前記インタフェースはメモリコントローラで実現されることを特徴とする請求項1に記載のシステム。

【請求項9】 前記メモリコントローラはNANDインタフェースロジックを含むことを特徴とする請求項8に記載のシステム。

【請求項10】 前記ブートストラップは前記ブートコードを貯蔵するための前記RAMを含むことを特徴とする請求項9に記載のシステム。

【請求項11】 前記RAMは前記ブートストラップの外部に位置することを特徴とする請求項9に記載のシステム。

【請求項12】 前記RAMは第1ローカルバスを通じて前記CPUコアに連結されたキャッシュメモリであることを特徴とする請求項1に記載のシステム。

【請求項13】 前記ブートストラップを前記キャッシュメモリに連結するためのラップをさらに含むことを特徴とする請求項12に記載のシステム。

【請求項14】 前記ブートストラップを前記キャッシュメモリに直接連結するための第2ローカルバスをさらに含むことを特徴とする請求項12に記載のシステム。

【請求項15】 前記インタフェースはメモリコントローラで実現されることを特徴とする請求項14に記載のシステム。

10 【請求項16】 前記メモリコントローラはNANDインタフェースロジックを含むことを特徴とする請求項14に記載のシステム。

【請求項17】 前記ブートストラップは前記ブートコードを貯蔵するための前記RAMを含み、前記RAMはローカルバスを通じて前記CPUコアに連結されたキャッシュメモリであることを特徴とする請求項1に記載のシステム。

20 【請求項18】 前記インタフェースとシステムバスを通じて前記NANDフラッシュメモリと前記システムを制御するためのメモリコントローラとをさらに含み、前記インタフェースは前記メモリコントローラと前記ブートストラップによってシステムバスが同時にアクセスされることを防止するための手段を含むことを特徴とする請求項1に記載のシステム。

【請求項19】 前記メモリコントローラはNANDインタフェースロジックを含むことを特徴とする請求項18に記載のシステム。

30 【請求項20】 CPUコア、システムメモリ、前記CPUコア及び前記システムメモリ間のデータ通信のためのインタフェースを有するシステムにおいて、少なくともSプログラムを貯蔵するNANDフラッシュメモリと、前記NANDフラッシュメモリに貯蔵されたデータが前記システムメモリにコピーされるようにするコードを貯蔵するRAMとを含むことを特徴とするシステム。

【請求項21】 前記OSプログラムは初期化時に、前記システムメモリにコピーされ、前記CPUコアは前記システムメモリにアクセスすることによって、前記OSプログラムを実行することを特徴とする請求項20に記載のシステム。

40 【請求項22】 前記インタフェースとシステムバスを通じて前記NANDフラッシュメモリと前記システムを制御するメモリコントローラをさらに含み、前記インタフェースは前記メモリコントローラ及びブートストラップによって前記システムバスが同時にアクセスされることを防止する手段を含むことを特徴とする請求項21に記載のシステム。

【請求項23】 前記メモリコントローラはNANDインタフェースロジックを含むことを特徴とする請求項22に記載のシステム。

【請求項 24】 CPUコア、システムメモリ、前記 CPUコア及び前記システムメモリ間のデータ通信のためのインタフェースを有するシステムにおいて、少なくともブートコードを貯蔵する NAND フラッシュメモリと、システムバスを通じて前記 NAND フラッシュメモリに連結され、前記 NAND フラッシュメモリから前記ブートコードを受け入れるブートストラップとを含み、前記ブートストラップは前記ブートコードを貯蔵するための RAM を含み、システムブーティング動作は前記 CPU コアによって前記 RAM から前記ブートコードを読み出す動作を含むことを特徴とする請求項 22 に記載のシステム。

【請求項 25】 前記インタフェースとシステムバスを通じて前記 NAND フラッシュメモリと前記システムを制御するメモリコントローラをさらに含み、前記インタフェースは前記メモリコントローラ及びブートストラップによって前記システムバスが同時にアクセスされることを防止する手段を含むことを特徴とする請求項 24 に記載のシステム。

【請求項 26】 前記メモリコントローラは NAND インタフェースロジックを含むことを特徴とする請求項 24 に記載のシステム。

【請求項 27】 CPU コア、システムメモリ、前記 CPU コア及び前記システムメモリ間のデータ通信のためのインタフェースを有するシステムにおいて、少なくとも OS プログラムを貯蔵する NAND フラッシュメモリと、

システム初期化信号の入力時に、前記 OS プログラムを前記システムメモリにコピーされるようにするコピーロジック回路とを含み、前記 CPU コアは前記システムメモリをアクセスすることによって、前記 OS プログラムを実行することを特徴とするシステム。

【請求項 28】 前記インタフェースとシステムバスを通じて前記 NAND フラッシュメモリと前記システムを制御するメモリコントローラをさらに含み、前記インタフェースは前記メモリコントローラ及びブートストラップによって前記システムバスが同時にアクセスされることを防止する手段を含むことを特徴とする請求項 27 に記載のシステム。

【請求項 29】 前記メモリコントローラは NAND インタフェースロジックを含むことを特徴とする請求項 28 に記載のシステム。

【請求項 30】 CPU コア、システムメモリ、前記 CPU コア及び前記システムメモリ間のデータ通信のためのインタフェースを有するシステムにおいて、少なくともブートコードを貯蔵するフラッシュメモリと、前記フラッシュメモリへの制御信号伝送を制御し、前記

インタフェースを通じて前記フラッシュメモリから前記ブートコードを受け入れるブートストラップと、前記フラッシュメモリを選択的に解鎖する第 1 及び第 2 メモリコントローラと、

フラッシュメモリの種類に従って前記第 1 及び第 2 メモリコントローラのうち一つの動作を選択する選択器とを含むことを特徴とするシステム。

【請求項 31】 前記フラッシュメモリは NOR 及び NAND フラッシュメモリのうち一つであることを特徴とする請求項 30 に記載のシステム。

【請求項 32】 前記第 1 メモリコントローラは NOR インタフェースロジックを含み、前記第 2 メモリコントローラは NAND インタフェースロジックを含むことを特徴とする請求項 31 に記載のシステム。

【請求項 33】 前記フラッシュメモリから入力された前記ブートコードを貯蔵する RAM をさらに含むことを特徴とする請求項 30 に記載のシステム。

【請求項 34】 前記選択器は前記第 1 及び第 2 メモリコントローラのうち一つの動作を選択するための選択ピンを含むことを特徴とする請求項 30 に記載のシステム。

【請求項 35】 CPU コア、システムメモリ及びメモリコントローラを有するコンピューティング装置を駆動する方法において、

NAND フラッシュメモリにブートコードを予め貯蔵する段階と、

システム初期化信号を受け入れる段階と、前記ブートコードを前記 NAND フラッシュメモリから RAM に伝達する段階と、

前記 CPU コアが前記 RAM に貯蔵された前記ブートコードを実行するようにする段階とを含むことを特徴とする方法。

【請求項 36】 前記システム初期化信号の入力時に、前記ブートコードを前記 NAND フラッシュメモリから前記 RAM に伝達する段階が完了されるまで前記 CPU コアの実行をホールドさせる段階をさらに含むことを特徴とする請求項 35 に記載の方法。

【請求項 37】 CPU コア、システムメモリ及びメモリコントローラを有するコンピューティング装置を駆動する方法において、

少なくとも一つの連結ピンに初期化パラミットを設定する段階と、

システム初期化信号を受け入れる段階と、NAND フラッシュメモリから前記システムメモリに OS プログラムを伝達する段階と、

前記 CPU コアが前記システムメモリからの前記 OS プログラムを実行するようにする段階とを含むことを特徴とする方法。

【請求項 38】 前記 OS プログラムを実行する以前に、前記少なくとも一つの連結ピンからの初期化パラミ

10

20

30

40

50

ットを利用して前記システムメモリ内のモードレジスタを設定する段階をさらに含むことを特徴とする請求項37に記載の方法。

【請求項39】 CPUコア、システムメモリ及びメモリコントローラを有するコンピューティング装置を駆動する方法において、
NANDフラッシュメモリ内にOSプログラムを予め貯蔵する段階と、

前記NANDフラッシュメモリからのデータ伝送を起こすための制御データを含むデータをROMに予め貯蔵する段階と、

システム初期化信号を受け入れる段階と、

前記ROMからの前記制御データを使用して前記NANDフラッシュメモリからのOSプログラムが前記システムメモリに伝達されるようにする段階と、

前記CPUコアが前記システムメモリからの前記OSプログラムを実行するようにする段階とを含むことを特徴とする方法。

【請求項40】 前記システムの初期化信号の入力時に、前記OSプログラムを前記NANDフラッシュメモリから前記RAMに伝送する段階が完了されるまで前記CPUコアの実行をホールドさせる段階をさらに含むことを特徴とする請求項39に記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はコンピューティング装置をブーティングするシステムに関するものであり、さらに具体的には、NANDフラッシュメモリ装置を利用して駆動されるシステム及びそのブーティング方法に関するものである。

【0002】

【従来の技術】PDAのような一般的なパーソナルコンピュータまたはコンピューティング装置において、バイオス(Basic Input/Output System: BIOS)に設けられたプログラムはパーソナルコンピュータまたはコンピューティング装置がONする時に実行される。バイオスプログラムを実行することによって、多い初期化機能が実行される。一般的に、そのような機能は、カスタムセッティング設定(custom settings)に対するCMOSセットアップをチェックする機能、インフラットハンドラ及び装置ドライバをローディングする機能、レジスタと装置管理を初期化する機能、ディスクドライブのように設置された構成要素または周辺装置に対するパワーオンセルフテスト(power-on-self-test: POST)を実行する機能、システム設定を表示する機能、どのような構成要素が駆動可能であるかを決定する機能、及びブートストラップシーケンスをスタートする機能を含む。通常、バイオス(またはブーティング)プログラムはROM、EPROMまたはNORフラッシュ

メモリに貯蔵される。

【0003】ブーティングプログラムがROMに貯蔵されれば、ROMが不揮発性であるので、貯蔵されたプログラムは変更することができない。貯蔵されたプログラムの細かい変更さえも、ROMの交換を必要にする。ブーティングプログラムがEPROMに貯蔵される場合に、貯蔵されたプログラムを変更するためには、以前に貯蔵されたプログラムが消去されなければならない。EPROM消去は別個の構成要素または装置をさらに要求する。ブーティングプログラムがROMまたはEPROMに貯蔵されれば、ブーティングプログラムに要求されるような変更または更新も容易には実行することができない。ブーティングプログラムがNORフラッシュメモリに貯蔵される場合に、貯蔵されたプログラムは消去、または更新することができる。しかし、NANDフラッシュメモリと比較すれば、NORフラッシュメモリは与えられたメモリ貯蔵容量に対して大きさがさらに大きく、製造の費用がより高い。

【0004】バイオスを貯蔵するNANDフラッシュメモリを有するシステムの一例が図1に示している。これはU.S. Patent No. 5,535,357に開示されている。図1を参照すれば、システム10はシステムバス17、NANDフラッシュメモリ18と内部インタフェースブロック15が含まれた複合チップ16及び複合チップ16とシステムメモリ19を制御するためのコントローラ11を含む。コントローラ11は中央処理装置CPUであり、これはコンピューティング機能を実行するCPUコア12、メモリコントローラ14及びコントローラ11内にある内部システムバス13を有する。メモリコントローラ14はNANDフラッシュメモリ18とシステムメモリ19との間のメモリマップを実行し、メモリマップに従って実行される機能をインタフェースするための内部インタフェースブロック15を使用する。内部インタフェースブロック15はレジスタまたはRAMのような貯蔵装置内のNANDフラッシュメモリのデータを一時的に貯蔵し、メモリコントローラ14の制御下にシステムバス17を通じて一時的に貯蔵されたデータをシステムメモリ19に伝達する。

【0005】内部インタフェースブロック15はNANDフラッシュメモリ18とインタフェースするためのNANDインタフェースロジック28とシステムバス17を通じてシステムメモリ19またはメモリコントローラ14とインタフェースするためのNORインタフェースロジック29を含む。NORインタフェースロジック29はNORフラッシュメモリとメモリコントローラ及び/またはシステムメモリの間でインタフェースするために、一般的に使用される回路である。フラッシュメモリがNORフラッシュメモリであれば、フラッシュメモリからの信号をNORインタフェース方式(当業者は“ROMインタフェース方式”という)に一致させるため

に、NANDインタフェースロジック28は不要である。NORインタフェース方式はバイトワード単位のアドレスに従って、メモリのランダムアクセスが可能なデータ伝送方式である。一方、NANDインタフェース方式によるデータ伝送は、ランダムアクセスではなく、ブロック単位のデータがブロックアドレス及びコマンドによって伝送される。

【0006】システム10はNANDインタフェース方式を利用したNANDインタフェースロジック28を通じてNANDフラッシュメモリ18のデータを内部インタフェースブロック15に伝達し、NORインタフェース方式を利用したNORインタフェースロジック29を通じて前記データをシステムメモリ19に伝達する。NANDフラッシュメモリ18からデータをアクセスするために、NANDインタフェース方式とNORインタフェース方式を経由しなければならぬので、データアクセス速度が低下する。さらに、そのような装置のシステム性能は最適化することができない。その理由は、メモリコントローラによってフラッシュメモリ内に貯蔵されたブーティングコードをアクセスするのに必要な時間がシステム性能の一つの測定要素になるからである。

【0007】さらに、NANDフラッシュメモリ18のすべてのデータが内部インタフェースブロック15にロードされなければならないし、NAND及びNORインタフェースを全部支援するロジック回路が要求されるので、内部インタフェースブロック15の大きさは大きくなければならない。したがって、図1に示したようなシステム10の場合は、費用が増加し、性能が低下する。

【0008】

【発明が解決しようとする課題】本発明の第1目的は、向上したシステムの性能を有するシステムを提供することである。

【0009】本発明の第2目的は、製造費用を低めることができるシステムを提供することである。

【0010】本発明の第3目的は、システムの性能を向上させることができるシステムブーティング方法を提供することである。

【0011】本発明の第4目的は、製造費用を低めることができるシステムブーティング方法を提供することである。

【0012】

【課題を解決するための手段】CPUコア、システムメモリ、前記CPUコア及び前記システムメモリ間のデータ通信のためのインタフェースを有するシステムは少なくともブートコードを貯蔵するNANDフラッシュメモリと、前記NANDフラッシュメモリへの制御信号伝送を制御し、前記NANDフラッシュメモリからのブートコードを前記インタフェースを通じて受け入れるブートストラップと、前記NANDフラッシュメモリからアクセスされた前記ブートコードを貯蔵するRAMとを含

み、システムブーティング動作は前記CPUコアによって前記RAMから前記ブートコードを読み出す動作を含む。望ましくは、前記ブートコードはシステム初期化プログラムとコピー命令プログラムとを含む。

【0013】本発明の一実施形態によれば、前記ブートストラップは前記NANDフラッシュメモリをインタフェースするためのNANDインタフェースロジックを含み、前記ブートコードを貯蔵するためのRAMは前記ブートストラップ内に集積される。

【0014】他の実施形態において、前記RAMは前記ブートストラップの外部に位置する。

【0015】前記システムは前記ブートコードが前記RAMに貯蔵されるまで前記CPUコアの動作を遅延させるディレーをさらに含む。前記ディレーは前記ブートストラップで実現され、前記インタフェースはメモリコントローラで実現される。前記メモリコントローラはNANDインタフェース方式で前記NANDフラッシュメモリをインタフェースするためのNANDインタフェースロジックを含む。

【0016】本発明の他の実施形態によれば、前記RAMは第1ローカルバスを通じて前記CPUコアに連結されたキャッシュメモリである。システムは前記ブートストラップを前記キャッシュメモリに連結するためのラップをさらに含む。第2ローカルバスは前記ブートストラップを前記キャッシュメモリに直接連結する。

【0017】前記システムは前記インタフェースとシステムバスを通じて前記NANDフラッシュメモリと前記システムを制御するためにメモリコントローラをさらに含む。前記インタフェースは前記メモリコントローラと前記ブートストラップによってシステムバスが同時にアクセスされることを防止するための手段を含む。

【0018】本発明の他の実施形態によれば、CPUコア、システムメモリ、前記CPUコア及び前記システムメモリ間のデータ通信のためのインタフェースを有するシステムは少なくともOSプログラムを貯蔵するNANDフラッシュメモリと、前記NANDフラッシュメモリに貯蔵されたデータが前記システムメモリにコピーされるようにするコードを貯蔵するRAMとを含む。望ましくは、前記OSプログラムは初期化時に、前記システムメモリにコピーされ、前記CPUコアは前記システムメモリをアクセスすることによって、前記OSプログラムを実行する。

【0019】また他の実施形態によれば、CPUコア、システムメモリ、前記CPUコア及び前記システムメモリ間のデータ通信のためのインタフェースを有するシステムは少なくともブートコードを貯蔵するNANDフラッシュメモリと、システムバスを通じて前記NANDフラッシュメモリに連結され、前記NANDフラッシュメモリから前記ブートコードを受け入れるブートストラップとを含み、前記ブートストラップは前記ブートコード

を貯蔵するためにRAMを含み、システムブーティング動作は前記CPUコアによって前記RAMから前記ブートコードを読み出す動作を含む。

【0020】本発明の他の特徴において、CPUコア、システムメモリ、前記CPUコア及び前記システムメモリ間のデータ通信のためのインタフェースを有するシステムは少なくともOSプログラムを貯蔵するNANDフラッシュメモリと、初期化パラメータを設定するための複数の連結ピンと、システム初期化信号の入力時に、前記OSプログラムをシステムメモリにコピーされるようにするコピーロック回路とを含み、前記CPUコアは前記システムメモリをアクセスすることによって、前記OSプログラムを実行し、システムブーティング動作は前記CPUコアによって前記RAMからブートコードを読み出す動作を含む。

【0021】本発明のまた他の特徴において、CPUコア、システムメモリ、前記CPUコア及びシステムメモリ間のデータ通信のためのインタフェースを有するシステムは少なくともブートコードを貯蔵するフラッシュメモリと、前記フラッシュメモリへの制御信号伝送を制御し、前記インタフェースを通じて前記フラッシュメモリから前記ブートコードを受け入れるブートストラップと、前記フラッシュメモリを選択的に制御する第1及び第2メモリコントローラと、フラッシュメモリの種類に従って前記第1及び第2メモリコントローラのうちの動作を選択する選択器とを含む。

【0022】前記フラッシュメモリはNOR及びNANDフラッシュメモリのうち一つである。前記第1メモリコントローラはNORインタフェースロジックを含み、前記第2メモリコントローラはNANDインタフェースロジックを含む。

【0023】前記システムは、前記フラッシュメモリから入力された前記ブートコードを貯蔵するRAMをさらに含むことが望ましい。前記選択器は前記第1及び第2メモリコントローラのうちの動作を選択するための選択ピンを含む。

【0024】CPUコア、システムメモリ、及びメモリコントローラを有するコンピューティング装置を駆動する方法は、NANDフラッシュメモリにブートコードを予め貯蔵する段階と、システム初期化信号を受け入れる段階と、前記ブートコードを前記NANDフラッシュメモリからRAMに伝達する段階と、前記CPUコアが前記RAMに貯蔵された前記ブートコードを実行させる段階とを含む。

【0025】このような駆動方法は、前記システム初期化信号の入力時に、前記ブートコードを前記NANDフラッシュメモリから前記RAMに伝達する段階が完了されるまで前記CPUコアの実行をホールドさせる段階をさらに含む。

【0026】

【発明の実施の形態】本発明の実施形態によるNANDフラッシュメモリを利用したシステムが図2に示されている。

【0027】図2を参照すれば、コンピューティングシステム20はコントローラ21、システムバス17、NANDフラッシュメモリ18及びシステムメモリ19を含む。コンピューティングシステム20はPDA、パーソナルコンピュータ、ラップトップ、パーソナルコンピュータまたはシステム初期化プログラム（一般的に、ブートコードという）によってブーティングされるようなシステムのコンピューティング装置の必須的な構成を含む。

【0028】コントローラ21はCPUコア12、内部システムバス13、メモリコントローラ14、ブートストラップ25及びインタフェース27を含む。コントローラ21は単一の半導体チップに実現することができ、一般的に、システムメモリ17に連結されたNANDフラッシュメモリ18とシステムメモリ19のようなどのようなメモリを制御して管理する。

【0029】システムバス17はコントローラ21、NANDフラッシュメモリ18及びシステムメモリ19のデータ伝送のために使用される。

【0030】NANDフラッシュメモリ18はシステム20をブーティングするためのブートコードを貯蔵し、オペレーティングシステム（以下、OSという）及び他のプログラムまたはデータをさらに貯蔵することができる。OSはMS-DOS（登録商標）またはWINDOWS（登録商標）である。これはブーティングまたは初期化過程の後に、システム20を動作するためのコントローラ21によって実行される。NANDフラッシュメモリ18に貯蔵されたデータは使用者の要求に基づいてユーザ貯蔵装置の構成コードと、どのように応用プログラムを利用するかを示すソフトウェアコードであり得る。

【0031】望ましくは、システムメモリ19はDRAMであり、これはデータ、命令及びそのようなことを貯蔵するメインメモリとして動作する。

【0032】CPUコア12はOSと応用及び動作プログラムを実行する。内部システムバス13はCPUコア12に、そしてそれからのデータを伝達し、メモリコントローラ14とブートストラップ25との間でデータを伝達する。

【0033】本発明の実施形態によれば、ブートコードは初期化の以前にNANDフラッシュメモリにまず貯蔵される。初期化動作によれば、NANDフラッシュメモリ18に貯蔵されたブートコードはブートストラップ25内の内部RAM26に伝達される。ブートコードの伝達完了された後に、システムの初期化コード及びコピー命令コードを含むブートコードはCPUコア12によって実行される。システム初期化コードが実行さ

れる時に、コントローラ21、システムメモリ19及び周辺装置が初期化される。コピーバック命令コードはOSまたは他の記憶されたデータがシステムメモリ19にロードされるようになる。OSまたは任意の時間にNANDフラッシュメモリから内部RAM26にロードされた他のデータの大きさはブートコードによって指定された値に従って決められるが、ブートストラップ25内のハードウェアロジックによって決定される。NOR及びNANDインタフェースロジック(図1参照)と比較すれば、ブートストラップ25が一つのNANDインタフェースロジックのみを含むので、ブートストラップ25は図1のインタフェースブロック15と比較して大きさが小さい。さらに、内部RAM26がブートコードのみを記憶するので、ブートストラップ25はNANDフラッシュメモリ18のすべてのデータを記憶するインタフェースブロック15と比較する時に、さらに小さい容量を有する。したがって、本発明によるシステム20の価格は図1のシステムのそれよりさらに低い。

【0034】メモリコントローラ14はメモリ動作を制御して管理し、メモリ動作はNANDフラッシュメモリ18に記憶されたOSまたはデータをシステムバス17を通じてシステムメモリ19に記憶する動作と、システムメモリ19からデータを読み出す動作を含む。そのような動作において、メモリコントローラ14はNANDフラッシュメモリ18とシステムメモリ19との間のメモリマッピング動作を実行する。

【0035】次に、図2に示したシステム20の動作を説明する。システム20がONする時に、ブートストラップ25はシステム初期化信号(例えば、パワーアップ信号とシステムリセット信号)を受け入れ、NANDフラッシュメモリ18に記憶されたブートコードをシステムメモリ19に伝達する。ブートコードが内部RAM26に伝達される間、ブートストラップ25はCPU12の動作をホールドするための制御信号を発生する。ブートコードの伝送が完了された後に、制御信号はリセットされ、CPUコア12は活性化されてポストブーティング(post-booting operation)動作を実行する。

【0036】また、CPUコア12の動作をホールドするために、コントローラ21内にディレー(図示せず)が使用される。例えば、システム初期化信号(例えば、パワーアップ信号とシステムリセット信号)はCPUコア12に連結されたブートストラップ25とディレーに同時に印加される。ブートコードがNANDフラッシュメモリ18から内部RAM26に伝達された後でシステム初期化信号の到達時間が遅延される。ディレーは遅延回路によって、またはソフトウェアによって実現される。ディレーはブートコード伝送に必要な時間と同一の時間だけ、またはそれより若干長く遅延させるように設定される。

【0037】CPUコア12の活性化によって、内部RAM26に記憶されたブートコードが実行される。ブートコード内のシステム初期化コードを実行することによって、システム20のハードウェアが初期化される。ブートコードのコピーバック命令コードを実行することによって、CPUコア12はNANDフラッシュメモリ内に記憶されたOSのようなプログラムまたは他のデータを読み出す。望ましくは、これはインタフェース27とブートストラップ25のNANDインタフェースロジック28を通じてページ単位に実行される。以後、CPUコア12は読み出されたデータまたはOSのようなプログラムをメモリコントローラ14とインタフェース27を通じてシステムメモリ19にコピーする。コピー動作が完了した後に、OSが実行される。したがって、ブーティング過程が完了した時に、システム20はシステムメモリ19からのOSによって駆動される。

【0038】本発明の実施形態によれば、NANDフラッシュメモリ18内に記憶されたブートコードを内部RAM26にコピーし、その次に、OSをシステムメモリ19に伝達することによって、実行されるブーティング動作は単一のNANDインタフェースを通じて行われる。図1のシステムで示したように、二段インタフェース方式(すなわち、NANDインタフェース方式とNORインタフェース方式)と比較する時に、ブーティング速度をより速く実現することができる。

【0039】NANDフラッシュメモリ内に記憶されたブートコードをブートストラップ25によって読み出し、読み出されたブートコードを内部RAM26に伝達する動作は図2及び図3に基づいて以下詳細に説明される。図3は本発明の望ましい実施形態によるNANDフラッシュメモリからブートコードを読み出す時のタイミング図を示す。

【0040】ブートストラップ25はシステム初期化信号に応答してNANDフラッシュメモリに記憶されたブートコードを読み出すための制御信号(例えば、CLE、ALE、CE、WE、RE及びR/B)を出力する。読み出し命令はアドレスad0、ad1、ad2と共に'00h'で生成される。システム初期化信号はパワーアップ信号(システムのパワーアップ時に生成される)、システムリセット信号、または再リセット信号を含む。

【0041】命令ラッチイネーブル信号CLEは所定の命令がNANDフラッシュメモリ18に入力される時に活性化される。アドレスラッチイネーブル信号ALEは所定のアドレスがNANDフラッシュメモリ18に入力される時に活性化される。

【0042】NANDフラッシュメモリ18は命令ラッチイネーブル信号CLE、チップイネーブル信号CE#、及び書き込みイネーブル信号WE#の活性化にตอบสนองしてシステムバス17を通じて読み出し命令00hを受

け入れる。ここで、“#”はアクティブロー状態を意味する。

【0043】NANDフラッシュメモリ18はアドレスラッチイネーブル信号ALE、チップイネーブル信号CE#及び書き込みイネーブル信号WE#の活性化にตอบสนองしてシステムバス17を通じてアドレスを受け入れる。生成されたアドレスの数はNANDフラッシュメモリアドレスステップ選択信号に従って設定される。

【0044】本発明の実施形態によれば、NANDフラッシュメモリ18の三段アドレス入力動作が図3に示されているが、アドレス入力過程が図3に示したことに制限されないことは、この分野の当業者に容易に理解することができる。NANDフラッシュメモリアドレスステップ選択信号に従って三段またはそれより多い段階から新たなアドレス入力過程が使用される。

【0045】本発明によれば、ブートストラップ25はNANDフラッシュメモリ18に貯蔵されたブートコードを読み出すための読み出し命令00hを発生し、その次に、アドレスad0、ad1、ad2を発生する。貯蔵されたブートコードは読み出し命令00hとアドレスad0、ad1、ad2に従ってページ単位に読み出される。読み出されたデータはNANDフラッシュメモリ18の内部バッファ（図示せず）に一時的に貯蔵される。

【0046】制御信号R/B#の活性化（‘ロー’）状態時に、データ（例えば、ブートコード）を内部バッファにコピーする動作が完了する。制御信号R/B#の非活性化（‘ハイ’）状態時に、データ（ブートコード：ここで、D0、D1、D2、D3）のコピー動作は完了しない。読み出しイネーブル信号RE#の活性化時に、内部バッファに貯蔵されたデータD0、D1、D2、D3はシステムバス17に伝送される。以後、システムバス17上のデータD0、D1、D2、D3は内部RAM26に伝送される。

【0047】本発明の望ましい実施形態によれば、ブートストラップ25はシステム初期化信号にตอบสนองしてCPUコア12の動作をホールド、または遅延させ、同時に、NANDインタフェースロジック28とインタフェース27を通じてNANDフラッシュメモリ18に制御信号CE#、CLE、ALE、WE#、RE#、R/B#を出力する。NANDフラッシュメモリ18から読み出されたデータはCE#、CLE、ALE、WE#、RE#及びR/B#のような制御信号にตอบสนองしてNANDインタフェースロジック28とインタフェース27を通じてブートストラップ25に伝送される。ブートストラップ25はブートコードを内部RAM26に貯蔵し、CPUコア12のホールド状態を解除する。または、CPUコア12は内部RAM26にブートコードを貯蔵するのに必要な時間よりさらに長い時間、遅延される。ディレーは調節可能なタイマである。したがって、CPUコア

12がホールド状態に維持される間、NANDフラッシュメモリ18に貯蔵されたブートコードは内部RAM26に伝送される。

【0048】以後、内部RAM26にロードされたブートコード内のシステム初期化コードを実行することによって、システム20のハードウェアが初期化される。ブートコード内のコピーループ命令コードを実行することによって、CPUコア2は、メモリコントローラ14とインタフェース27を通じてNANDフラッシュメモリ18からシステムメモリ19にOSが伝送されるようにすることが望ましい。インタフェース27内のマルチプレクシングまたはデータ選択回路（図示せず）はメモリバス17へのアクセスをマルチプレクシングし、その結果、メモリコントローラ14とブートストラップ25が同時にシステムバス17をアクセスすることを防止することができる。

【0049】システムメモリ19へのOS伝送が完了された後に、システムメモリ19からのOSが実行され、システム20はOSによって駆動される。

【0050】本発明によるNANDフラッシュメモリを利用したブートシステムの第2実施形態が図4に示されている。この実施形態によれば、内部RAM33はブートストラップ32の外部に配置される。ブートストラップ32から内部RAM33へのアクセスは内部システムバス13を通じて行われることが望ましい。それと異なり、NANDフラッシュメモリ18からのデータ伝送動作は先の説明と同一である。

【0051】本発明によるNANDフラッシュメモリを利用したブートシステムの第3実施形態が図5に示されている。この実施形態によれば、ブートストラップ32から内部RAM33へのアクセスはバス34を通じて行われることが望ましい。これと異なり、NANDフラッシュメモリ18からのデータ伝送動作は先の説明と同一である。内部RAM34へのローディング動作のために、バス34を使用することによって、NANDフラッシュメモリ18からブートコードを伝送するのに必要な時間が減り、その結果、システム性能が向上することができる。

【0052】本発明によるNANDフラッシュメモリを利用したブートシステムの第4実施形態が図6に示されている。図6のシステム50はコントローラ51、NANDフラッシュメモリ18、システムバス17及びシステムメモリ19を含む。

【0053】コントローラ51はブートストラップ52、CPUコア12、内部システムバス13、内部RAM及びメモリコントローラ54を含み、メモリコントローラ54はNANDインタフェースロジック28を有する。図6に示したように、内部RAM53はブートストラップ52内に位置するが、図4及び図5に示したように、ブートストラップ52の外部に配置される。この実

施形態によれば、メモリコントローラ54はシステムメモリ19を制御し、NANDフラッシュメモリ18を直接アクセスするの用に使用される。したがって、この実施形態によるメモリコントローラ54は図2に示したインタフェース27によって、以前に実行した機能を実行することができる。

【0054】ブートストラップ52はシステム初期化信号にตอบสนองしてCPUコア12の動作をホールドし、同時に、NANDインタフェースロジック28によるNANDインタフェース方式を使用して内部システムバス13を通じてNANDフラッシュメモリ18に貯蔵されたブートコードを読み出す。以後、ブートストラップ52は読み出されたブートコードを内部RAM53にロードする。システム初期化信号はパワーオン信号またはリセット信号にตอบสนองして生成された信号である。

【0055】ブートコード内のシステム初期化コードを実行することによって、システム20のハードウェアが初期化される。ブートコード内のコピーバック命令コードを実行することによって、CPUコア12はメモリコントローラ54内に貯蔵されたOSを読み出し、そのOSをシステムメモリ19にロードする。システムメモリ19へのOS伝送が完了された後に、システムメモリ19からのOSが実行される。

【0056】本発明によるNANDフラッシュメモリを利用したブートシステムの第5実施形態が図7に示されている。コントローラ61はCPUコア12、ブートストラップ62、内部システムバス13、メモリコントローラ54及び内部RAM63を含む。内部RAM63がブートストラップ62の外部に配置されている。

【0057】ブートストラップ62はシステム初期化信号にตอบสนองしてCPUコア12の動作をホールドし、同時にメモリコントローラ54のNANDインタフェースロジック28によるNANDインタフェース方式を利用してNANDフラッシュメモリ18内に貯蔵されたブートコードを読み出す。以後、ブートストラップ62は内部システムバス13を通じて内部RAM63に読み出されたブートコードをロードする。

【0058】本発明によるNANDフラッシュメモリを利用したブートシステムの第6実施形態が図8に示されている。コントローラ71はCPUコア12、ローカルバス75、キャッシュメモリ73、ラップ72、内部システムバス13、メモリコントローラ54及びブートストラップ74を含む。ラップ72はソフトウェアプログラムまたはハードウェアであることが望ましい。これはキャッシュメモリ73への、またはそれからのアクセスを選択的に制御するために実行される。キャッシュメモリ73は種々使用されたデータを一時的に貯蔵するためにCPUコア12に隣接して配置される。キャッシュメモリ73のアクセス時間は図7の内部RAMのそれよりさらに短い。

【0059】キャッシュメモリ73が内部システムバス13を通じてアクセスされる場合に、ブートストラップ74はシステム初期化信号にตอบสนองしてCPUコア12の動作を遅延、またはホールドし、同時に、NANDインタフェースロジック28によるNANDインタフェース方式を利用してNANDフラッシュメモリ18内に貯蔵されたブートコードを読み出す。以後、ブートストラップ74は内部システムバス13を通じてキャッシュメモリ73に読み出されたブートコードを書き込む。

【0060】キャッシュメモリ73がローカルバス75を通じてアクセスされる場合に、ブートストラップ74はシステム初期化信号にตอบสนองしてCPUコア12の動作をホールドし、同時に、NANDインタフェースロジック28を利用してNANDフラッシュメモリ18内に貯蔵されたブートコードを読み出す。以後、ブートストラップ74はラップ(wrapper)72及びローカルバス75を通じてキャッシュメモリ73内に読み出されたブートコードを書き込む。本発明によるブートシステムを利用したシステムブーティング動作はより速く行われ、その理由は、キャッシュメモリ73のアクセス時間がさらに短いからである。

【0061】本発明によるNANDフラッシュメモリを利用したブートシステムの第7実施形態が図9に示されている。コントローラ81はキャッシュメモリ73とブートストラップ74との間でデータを伝達するための第2ローカルバス82を含む。図9のブートストラップ74は図8のブートストラップと同一の方式で動作する。

【0062】ブートストラップ74はシステム初期化信号にตอบสนองしてCPUコア12の動作をホールドし、同時に、メモリコントローラ54のNANDインタフェースロジック28を利用してNANDフラッシュメモリ18内に貯蔵されたブートコードを読み出す。以後、ブートストラップ74はバス82を通じてキャッシュメモリ73内に読み出されたブートコードを書き込み(コピー、ロード及び伝送)する。キャッシュメモリ73にブートコードを貯蔵した後に、次の動作はバス82が使用されることを除いては図6の説明と同一である。

【0063】本発明によるNANDフラッシュメモリを利用したブートシステムの第8実施形態が図10に示されている。コントローラ91はCPUコア12、ローカルバス75、キャッシュメモリ73、ラップ72、内部システムバス13、メモリコントローラ54及びキャッシュ及びブートストラップ92を含む。キャッシュ及びブートストラップ92は内蔵されたキャッシュメモリを有するブートストラップである。

【0064】キャッシュ及びブートストラップ92はシステム初期化信号にตอบสนองしてCPUコア12の動作をホールドし、同時に、NANDフラッシュメモリ18内に貯蔵されたブートコードを読み出す。以後、キャッシュ及びブートストラップ92は内部システムバス13を通

じてストラップ9 2内のキャッシュメモリに読み出されたブートコードを書き込む。また、ストラップ7 2はローカルバス7 5を通じてブートコードをロードするように設定される。

【0065】本発明によるNANDフラッシュメモリを利用したブートシステムの第9実施形態が図11に示されている。コントローラ101はCPUコア12、メモリコントローラ54、コピーロジックブロック105、内部システムバス13及びオプションピン106を含む。オプションピンは電源電圧VCCまたは接地電圧GNDに選択的に連結される。本発明の実施形態によれば、コントローラ101はシステムメモリ19内のモードレジスタMRS（図示せず）を設定することによって、システムメモリが初期化される初期化動作を実行する。MRSはオプションピン106を使用して設定される。MRSは使用されるシステムメモリ19の動作モード（例えば、CASレイタンスまたはバスの長さ）に予め設定される。例えば、NANDフラッシュメモリ内で一回にシステムメモリ19にロードされるデータの大きさはバスの長さによって決められる。コピーロジックブロック105はOSのようなデータまたはNANDフラッシュメモリ18に貯蔵された一般的なデータをシステムメモリ19にコピーする動作を指示し、コントローラ101で使用される一連の命令を含む。この実施形態によれば、ブートコードは不要であり、NANDフラッシュメモリ18に貯蔵される必要がない。CPUコア12の動作は、システム初期化信号に反応してホールドされる。同時に、コピーロジックブロック105はOSのようなデータとNANDフラッシュメモリ18で読み出されたデータをシステムメモリ19に貯蔵するように設定される。コピー動作が完了された後に、CPUコア12は活性化され、システムメモリ19内のOSが実行される。システム20はその次にOSに反応して駆動される。

【0066】そのような装置100において、システムメモリ19を初期化するためのシステム初期化動作は、オプションピン106の連絡情報に従って完了される。ブートコードをコントローラにコピーする動作が除去されたので、システムブーティング速度が向上する。

【0067】本発明によるNANDフラッシュメモリを利用したブートシステムの第10実施形態が図12に示されている。コントローラ110はCPUコア12、メモリコントローラ54、内部システムバス13及びROMブロック115を含む。ここで、ROMブロック115は図11のコピーロジックブロック105を代替する。システム初期化コード及びNANDフラッシュメモリ18に貯蔵されたデータがシステムメモリ19にコピーされるようにする命令を含むROMデータはROMブロック115に予め貯蔵される。ROMブロック115はマスクROM、フラッシュメモリまたはそのようなも

のを含むものが望ましい。

【0068】システム初期化信号に反応して、CPUコア12は活性化され、ROMブロック115に含まれたシステム初期化コードに従って初期化動作を実行し、NANDフラッシュメモリ18に貯蔵されたOSまたは一般的なデータをシステムメモリ19にコピーする。コピー動作が完了された後に、CPUコア12は活性化され、システムメモリ19内のOSが実行される。すなわち、ブーティング動作が終了される時に、システム110はOSに反応して駆動される。

【0069】本発明によるNANDフラッシュメモリを利用したブートシステムの第11実施形態が図13に示されている。システム120はコントローラ121、ブートストラップ25、NANDフラッシュメモリ18及びシステムメモリ19を含む。コントローラ121はCPUコア12、内部システムバス13及びメモリコントローラ14を有する。本発明によれば、ブートストラップ25はコントローラ121の外部に配置され、システムバス17を通じてコントローラ121に選択的に連結される。

【0070】CPUコア12の動作は、システム初期化信号に反応してホールドされ、ブートストラップ25は同時にNANDフラッシュメモリ18内のブートコードをシステムバス17を通じてRAM26にロードする。ブートコードを内部RAM26に貯蔵した後に、次の動作は図2で説明したものと同一である。

【0071】本発明によるNANDフラッシュメモリを利用したブートシステムの第12実施形態が図14に示されている。システム130はコントローラ131、フラッシュメモリ135、システムメモリ19及びシステムバス17を含む。コントローラ131はCPUコア12、NORインタフェースロジック29を有する第1メモリコントローラ133、NANDインタフェースロジックを有する第2メモリコントローラ134、選択回路136及びオプションピン132を有する。第1及び第2メモリコントローラ133、134はフラッシュメモリ135の種類に従って選択回路136によってシステムバス17に選択的に連結される。オプションピン132は電源電圧VCまたは接地電圧GNDに連結される。第1メモリコントローラ133または第2メモリコントローラ134はオプションピン132が電源電圧VCCまたは接地電圧GNDに連結されたか否かを知らせる連絡情報に従って選択的に活性化される。

【0072】本発明によれば、フラッシュメモリ135はNORフラッシュメモリ、NANDフラッシュメモリまたは他の形態のメモリである。NORフラッシュメモリがフラッシュメモリ135として使用されれば、第1メモリコントローラ133はNORインタフェースロジック29とシステムバス17を通じてNORフラッシュメモリに連結される。この場合に、第1メモリコント

ーラ 133 は第 1 メモリコントローラ 133 内の一般的な NOR インタフェース方式を使用してシステムバス 17 を NOR フラッシュメモリと相互連結する。したがって、システム 130 のコントローラ 131 は NOR フラッシュメモリのデータ (例えば、ブートコード、OS 及び一般的なデータ) を容易にアクセスすることができる。NAND フラッシュメモリがフラッシュメモリ 135 として使用されれば、第 2 メモリコントローラ 134 は NAND インタフェース方式をロジック 28 とシステムバス 17 を通じて NAND フラッシュメモリに連結される。この場合に、第 2 メモリコントローラ 134 は NAND インタフェース方式を利用して、図 2 乃至図 13 で説明した過程に従って、システムバス 17 を NAND フラッシュメモリと相互連結する。結果的に、システム 130 のコントローラ 131 はフラッシュメモリ 135 の種類に関係なしに、データ (例えば、ブートコード、OS 及び一般的なデータ) をアクセスすることができる。

【0073】本発明によるフラッシュメモリ内に貯蔵されたブートコードをローディングする手順を示すフローチャートが図 15 に示されている。図 2 乃至図 13 を参照すれば、各図面に示したブートストップの初期状態は遊休状態 (idle state) である (150 段階)。

【0074】システム初期化信号に応じて、ブートストップはフラッシュメモリのページの大きさ、密度及びデータの幅を検出し、同時に、システムの CPU をホールドするために制断信号を出力する。または、システムはブートコードが内部 RAM にロードされるまで CPU をホールドさせるためのディレーを使用する (151 段階)。

【0075】ブートストップは図 3 で説明した方法に従って、フラッシュメモリ内に貯蔵されたブートコードをページ段位に読み出し、読み出されたデータを所定のバスまたはインタフェースを通じて、コントローラの内部に、または外部に位置した内部 RAM にロードする (152 段階)。

【0076】153 段階、ブートストップはロードされたブートコードの伝送大きさを検出する。伝送の大きさが '0' であれば、手順は 152 段階に戻す。

【0077】伝送の大きさが '0' であれば、すなわち、フラッシュメモリに貯蔵されたブートコードが全部内部 RAM にロードされれば、ブートストップはブートコードをロードすることを終了し、システムの CPU を活性化させる (154 段階)。

【0078】システムブーティング方法によると、フラッシュメモリに貯蔵されたブートコードは CPU コアが動作する前に、コントローラ内に、または外部に位置した内部 RAM にロードされる。内部 RAM に貯蔵されたブートコードによって、フラッシュメモリに貯蔵された所定の OS が外部システムメモリ 19 にコピーされる。

コピー動作が完了された後に、システムメモリ 19 内の OS が実行される。

【0079】OS と関連したすべてのプログラム、ファイルまたはデータがシステムメモリ 19 にロードされた後に、システムはシステムメモリ 19 内の OS によって駆動される。

【0080】以上、本発明による回路の構成及び動作を上述の説明及び図面に従って示したが、これは例を挙げて説明したことに過ぎず、本発明の技術的思想及び範囲を逸脱しない範囲内で多様な変化及び変更が可能なこととはもちろんである。

【0081】

【発明の効果】本発明の実施形態によれば、NAND フラッシュメモリ 18 内に貯蔵されたブートコードを内部 RAM 26 にコピーし、その次に、OS をシステムメモリ 19 に伝送することによって実行されるブーティング動作は単一の NAND インタフェースを通じて行われる。図 1 のシステムで見られたように、二段インタフェース方式すなわち、NAND インタフェース方式と NOR インタフェース方式) と比較する時に、本発明によるシステムのブーティング速度をより速く実現することができる。

【図面の簡単な説明】

【図 1】 一般的な NAND フラッシュメモリを有するコンピューティングシステムを示す図面。

【図 2】 本発明の望ましい実施形態による NAND フラッシュメモリを利用したシステムを示す図面。

【図 3】 フラッシュメモリからのブートコードを伝送する過程を説明するためのタイミング図。

【図 4】 本発明によるコンピューティングシステムの他の実施形態。

【図 5】 本発明によるコンピューティングシステムのまた他の実施形態。

【図 6】 本発明によるコンピューティングシステムのまた他の実施形態。

【図 7】 本発明によるコンピューティングシステムのまた他の実施形態。

【図 8】 本発明によるコンピューティングシステムのまた他の実施形態。

【図 9】 本発明によるコンピューティングシステムのまた他の実施形態。

【図 10】 本発明によるコンピューティングシステムのまた他の実施形態。

【図 11】 本発明によるコンピューティングシステムのまた他の実施形態。

【図 12】 本発明によるコンピューティングシステムのまた他の実施形態。

【図 13】 本発明によるコンピューティングシステムのまた他の実施形態。

【図 14】 本発明によるコンピューティングシステム

21

22

のまた他の実施形態。

【図15】 本発明によるフラッシュメモリ内に貯蔵されたブートコードをローディングする手順を示すフローチャートである。

【符号の説明】

12 CPUコア

13 内部システムバス

14 メモリコントローラ

17 システムバス

18 NANDフラッシュメモリ

19 システムメモリ

20 システム

21 コントローラ

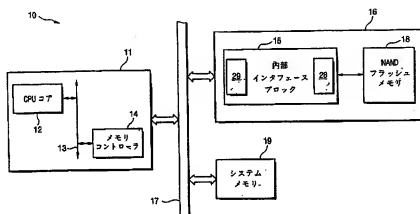
25 ブートストラップ

26 内部RAM

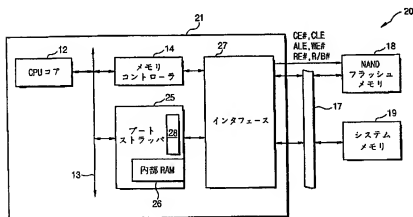
27 インタフェース

28 NANDインタフェースブロック

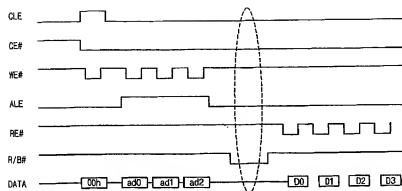
【図1】



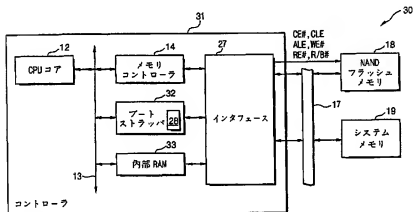
【図2】



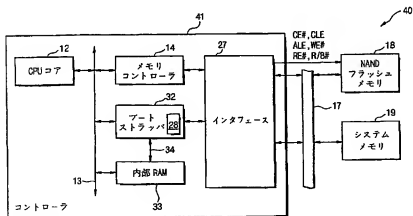
【図 3】



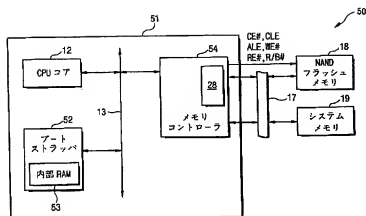
【図 4】



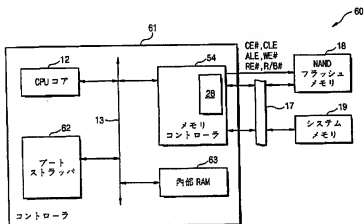
【図 5】



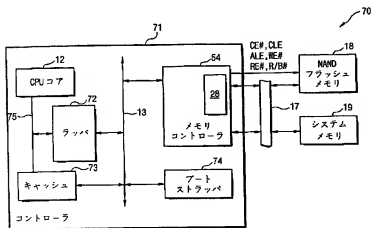
【図 6】



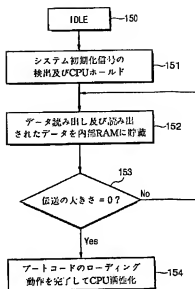
【図 7】



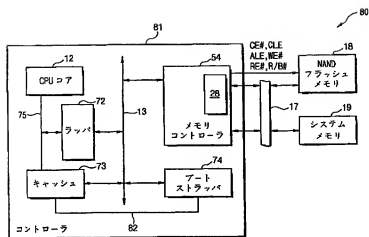
【図 8】



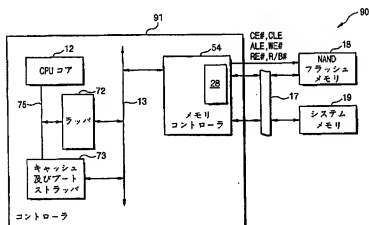
【図 15】



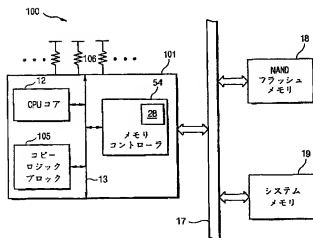
【圖9】



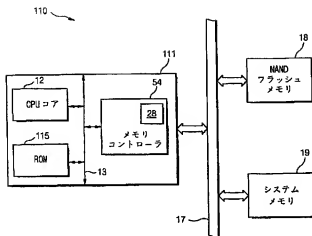
【图 10】



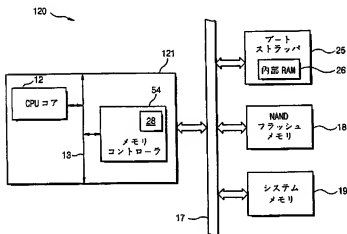
【圖 1 1】



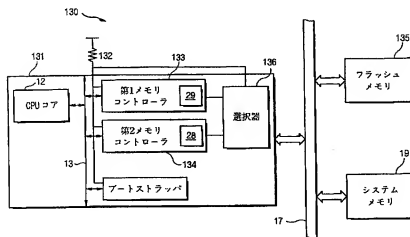
【図 1 2】



【図 1 3】



【図 1 4】



フロントページの続き

(72)発明者 崔 永準

大韓民国京畿道城南市盆唐区金谷洞142番

地 チョンソルタウン814-1802

(72)発明者 権 錫千

大韓民国京畿道城南市盆唐区亭子洞122

サンロクタウンウォゾンアパートメント
325棟801号

Fターム(参考) 5B060 AA18 MM02

5B076 AA01 BB12